UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/823,155 | 03/29/2001 | Daniel R. Gaur | 42390P10774 | 8114 |

8791 7590 02/07/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

| EXAMINER | |
|---|---|
| ZHEN, LI B | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2194 | |

DATE MAILED: 02/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/823,155 | GAUR ET AL. | |
| | Examiner | Art Unit | |
| | Li B. Zhen | 2194 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *07 November 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1 – 35 are pending in the current application.

### *Claim Rejections - 35 USC § 103*

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. **Claims 1, 2, 5 – 8, 10 – 16, 18 – 21, 24 – 27 and 29 – 35 rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,772,189 to Asselin in view of U.S. Patent No. 6,470,397 to Shah et al. [hereinafter referred to as Shah], both references cited in the previous office action.**

4. As to claim 1, Asselin teaches the invention substantially as claimed including a computer-implemented method comprising:

requesting a first deferred procedure call for a first interrupt event [ISR queues a work item on a selected work list and then queues a selected DPC by calling KeInsertQueueDpc( ) with a parameter corresponding to a DPC object that has been created; col. 5, lines 50 – 67];

requesting at least one other deferred procedure call for a second interrupt event associated with the source [When 4 the counter is incremented from 7 to 8, after using WorkList[7], the next work item will be placed on WorkList[0] again and so forth. When the counter rolls over, the WorkList[7] to WorkList[0] sequence will be maintained, since the number of work lists in the useable subset of DPC objects and associated work lists is a power of two; col. 6, line 59 – col. 7, line 19], wherein the first interrupt event comprises one type [DPC routine handles requests, command completions, and external communication responses for all of the devices supported by the interrupt; col. 2, lines 38 – 50] of event and the second interrupt event comprises another type of

event [interrupt service routines in device drivers also have a DPC (Deferred Procedure Call) object and work list associated with the interrupt; col. 5, lines 38 – 50];

assigning the first deferred procedure call and the at least one other deferred procedure call to a resource [associated DPC object is queued on the system DPC queue (step 36)...queuing is repeated in a round-robin fashion for every work item that needs to be performed; col. 7, lines 27 – 35];

processing the first interrupt event with the first deferred procedure call [kernel removes the DPC object from the queue (step 40), and calls the DPC routine associated with the DPC object (step 42); col. 7, lines 43 – 60]; and

processing the second interrupt event with the at least one other deferred procedure call [DPC routine removes and processes each work item from the work list, leaving the work list empty (step 44 and decision 45); col. 7, lines 43 – 60].

5. Although Asselin teaches the invention substantially as claimed, Asselin does not specifically teach processing each interrupt event with a different deferred procedure call.

6. However, Shah teaches a driver architecture supported by a variety of routines, including a interrupt DPC routine [col. 9, lines 16 – 23] and processing each interrupt event [with a different deferred procedure call [send DPC routine is scheduled by the Send Callback routine upon the stoppage of the send queue resulting from a shortage of send buffers....receive DPC routine is scheduled by the Receive Callback routine after queuing an incoming data packet; col. 15, lines 5 – 38].

7. It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of processing each interrupt event with a different deferred procedure call as taught by Shah to the invention of Asselin because this provides a system for running network and storage protocols over the same communication link, reduces the number of interface cards, interface circuits, and communication links, which provides for reduced costs and smaller system as compared with conventional systems [col. 3, line 65 – col. 4, line 7 of Shah].

8. As to claim 2, Asselin teaches assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a processor exhibiting a single thread of execution; and executing the first deferred procedure call and the at least one other deferred procedure call on the single thread ["lock" the queued DPC requests associated with a DPC object for execution on a single processor; col. 2, lines 50 – 60].

9. As to claim 5, Asselin teaches assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a multi-processor system [multiprocessor system may be executing an operating system providing interrupt handling services, and the DPC may be queued by executing a call to a service provided by the operating system for queueing DPC's; col. 3, lines 47 – 56]; and executing the first deferred procedure call on one processor of the multi-processor system while executing the at least one other deferred procedure call on another processor of the multi-processor system [multiple processors can be handing the command completion tasks instead of just one processor executing a single DPC routine and working on a single work list of work items; col. 7, lines 19 – 43].

10. As to claim 6, Asselin teaches assigning the first deferred procedure call to a resource comprising a first processor [first work item will be queued on WorkList[0]; col. 7, lines 1 – 19]; assigning the at least one other deferred procedure call to a resource comprising a second processor [proceeding using WorkList[ counter value & mask ] where mask is 7 and & is the logical AND operator and WorkList[n] indicates the n.sup.th member of an array of work lists; col. 7, lines 1 – 19]; and executing the first deferred procedure call on the first processor while executing the at least one other deferred procedure call on the second processor [DPC objects created in step 22 has a selectable subset large enough for each processor in the system to be executing the DPC routine independently for each DPC object and work list; col. 7, lines 43 – 60].

11. As to claim 7, Asselin teaches a third interrupt event associated with the source with the first deferred procedure call, the third interrupt event comprising a third type of event [DPC routine handles requests, command completions, and external communication responses for all of the devices supported by the interrupt; col. 2, lines 38 – 50].

12. As to claim 8, Asselin as modified teaches a computer implemented method comprising:

requesting a first deferred procedure call for a first interrupt event [ISR queues a work item on a selected work list and then queues a selected DPC by calling KeInsertQueueDpc( ) with a parameter corresponding to a DPC object that has been created; col. 5, lines 50 – 67 of Asselin];

requesting at least one other different deferred procedure call [col. 15, lines 5 – 38 of Shah] for a second interrupt event associated with the source [When 4 the counter is incremented from 7 to 8, after using WorkList[7], the next work item will be placed on WorkList[0] again and so forth; col. 6, line 59 – col. 7, line 19 of Asselin], wherein the first interrupt event comprises one type of event [DPC routine handles requests, command completions, and external communication responses for all of the devices supported by the interrupt; col. 2, lines 38 – 50 of Asselin] and the second interrupt event comprises another type of event [interrupt service routines in device drivers also have a DPC (Deferred Procedure Call) object and work list associated with the interrupt; col. 5, lines 38 – 50 of Asselin]; and

processing the first interrupt event with the first deferred procedure call [kernel removes the DPC object from the queue (step 40), and calls the DPC routine associated with the DPC object (step 42); col. 7, lines 43 – 60 of Asselin] while processing the second interrupt event with the at least one other deferred procedure call [DPC routine removes and processes each work item from the work list, leaving the work list empty (step 44 and decision 45); col. 7, lines 43 – 60 of Asselin].

13. As to claim 10, Asselin teaches executing the first deferred procedure call on a first processor; and executing the at least one other deferred procedure call on a second processor [DPC objects created in step 22 has a selectable subset large enough for each processor in the system to be executing the DPC routine independently for each DPC object and work list; col. 7, lines 43 – 60].

14. As to claim 11, this is rejected for the same reasons as claim 7 above.

15. As to claim 12, Asselin as modified teaches a driver [device driver 64, Fig. 3; col. 5, lines 9 – 50 of Asselin] comprising:

an interrupt handler [interrupt handling services; col. 3, lines 47 – 56 of Asselin] to identify interrupt events associated with a source [executes an ISR (Interrupt Service Routine) in response to an interrupt from a device; col. 3, lines 26 – 33 of Asselin]; and

a first deferred procedure call, the first deferred procedure call to process a first type of the interrupt events [ISR queues a work item on a selected work list and then queues a selected DPC by calling KeInsertQueueDpc( ) with a parameter corresponding to a DPC object that has been created; col. 5, lines 50 – 67 of Asselin]; and

a second different deferred procedure call [col. 15, lines 5 – 38 of Shah], second deferred procedure call to process a second type of the interrupt events [When 4 the counter is incremented from 7 to 8, after using WorkList[7], the next work item will be placed on WorkList[0] again and so forth. When the counter rolls over, the WorkList[7] to WorkList[0] sequence will be maintained, since the number of work lists in the useable subset of DPC objects and associated work lists is a power of two; col. 6, line 59 – col. 7, line 19 of Asselin].

16. As to claim 13, Asselin teaches assigning the first and second deferred procedure calls to a resource for execution [associated DPC object is queued on the system DPC queue (step 36)...queuing is repeated in a round-robin fashion for every work item that needs to be performed; col. 7, lines 27 – 35].

17. As to claim 14, Asselin teaches assigning the first deferred procedure call to a first resource for execution and the second deferred procedure call to a second resource for execution [multiple processors can be handing the command completion tasks instead of just one processor executing a single DPC routine and working on a single work list of work items; col. 7, lines 19 – 43].

18. As to claim 15, Asselin as modified teaches a computer system comprising:

a driver [device driver 64, Fig. 3; col. 5, lines 9 – 50 of Asselin] stored in a memory of the computer system, the driver including an interrupt handler [interrupt handling services; col. 3, lines 47 – 56 of Asselin] to identify interrupt events [col. 3, lines 26 – 33 of Asselin];

a first deferred procedure call, the first deferred procedure call to process a first type of the interrupt events [col. 5, lines 50 – 67 of Asselin]; and

a second different deferred procedure call [col. 15, lines 5 – 38 of Shah], second deferred procedure call to process a second type of the interrupt events [col. 6, line 59 – col. 7, line 19 of Asselin]; and

a processor to execute the two deferred procedure calls [col. 7, lines 19 – 43 of Asselin].

19. As to claim 16, Asselin teaches the interrupt handler to assign the two deferred procedure calls to a single thread exhibited by the processor for execution [col. 2, lines 50 – 60].

20. As to claim 18, Asselin teaches assign the first deferred procedure call to the processor and the second deferred procedure call to a second processor for execution [col. 7, lines 19 – 43].

21. As to claim 19, Asselin teaches at least one peripheral device, the interrupt events associated with the at least one peripheral device [col. 5, lines 9 – 49].

22. As to claims 20, 21, 24 – 27, 29 and 30, these are system claims that correspond to method claims 1, 2, 5 – 8, 10 and 11; note the rejections to claims 1, 2, 5 – 8, 10 and 11 above, which also meet these system claims.

23. As to claims 31 – 35, Asselin teaches the source comprises a peripheral device of a computer system [col. 5, lines 9 – 49].

24. **Claims 3, 4, 9, 17, 22, 23 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Asselin and Shah further in view of U.S. Patent No. 6,378,004 to Galloway et al. [hereinafter Galloway, cited in the previous office action].**

25. As to claim 3, Asselin as modified teaches processing deferred procedure call in a multiprocessor system [col. 7, lines 19 – 43 of Asselin], but does not teach a plurality of threads and executing the deferred procedure calls with threads.

However, Galloway teaches assigning the first deferred procedure call and the at least one other deferred procedure call [a deferred procedure call (DPC) is used to set an event object; col. 13, line 52 - col. 14, line 9] to a resource comprising a processor exhibiting a plurality of threads [a multitasking operating system (not shown) that supports multiple threads of execution within a running process and a kernel for handling thread management; col. 3, lines 11 - 33]; and executing the first deferred procedure call on one thread of the plurality of threads while executing the at least one other deferred procedure call on another thread of the plurality of threads [thread can execute any part of an application's code, including a part currently being executed by another thread; col. 3, lines 10 – 33].

26. It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of assigning the first deferred procedure call and the at least one other deferred procedure call to a resource comprising a processor exhibiting a plurality of threads and executing the deferred procedure calls with threads as taught by Galloway to the invention of Asselin as modified because this divides the

available CPU time among the threads that need it, utilize preemptive multitasking for allocating small slices of CPU time among the competing threads and provides the ability to have several applications open and working at the same time [col. 3, lines 25 - 30 of Galloway].

27. As to claim 4, Asselin as modified teaches assigning the first deferred procedure call to a resource comprising a first thread of a processor and assigning the at least one other deferred procedure call to a resource comprising a second thread of the processor [col. 8, lines 1 – 13 of Asselin]; and executing the first deferred procedure call on the first thread while executing the at least one other deferred procedure call on the second thread [col. 3, lines 10 – 33 of Galloway].

28. As to claims 9 and 17, Asselin as modified teaches executing the first deferred procedure call on a first thread of a processor [col. 8, lines 1 – 13 of Asselin]; and executing the at least one other deferred procedure call on a second thread of the processor [col. 3, lines 10 – 33 of Galloway].

29. As to claims 22, 23 and 28, these are system claims that correspond to method claims 3, 4 and 9; note the rejection to claims 3, 4 and 9 above, which also meet these system claims.

***Response to Arguments***

30. Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection. In the response dated 11/07/2005, applicant argued that Shah does not disclose the use of different interrupt DPCs to process alternate types of interrupt events but, rather, Shah discloses using the same interrupt DPC to handle a number of different tasks [p. 17, lines 3 – 12]. Examiner notes that the previously cited portion of Shah does not disclose the use of different interrupt DPCs to process alternate types of interrupt events. However, Shah discloses using different interrupt DPC to handle different types events on col. 15, lines 5 – 38. In

particular, Shah discloses scheduling a send DPC routine to handle the event of stoppage of the send queue resulting from a shortage of send buffers [col. 15, lines 5 – 15] and a receive DPC routine to handle the event of the queuing of an incoming data packet [col. 15, lines 29 – 38]. Therefore, Shah teaches using different DPCs [send and receive DPC] to handle different events and the claims remained rejected over the combination of Asselin and Shah and the combination of Asselin, Shah and Galloway.

## CONTACT INFORMATION

31. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on 571-272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen
Examiner
Art Unit 2194

lbz

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER